WiMatch: Wireless Resource Matchmaking

Kirk Webb
School of Computing
University of Utah
Salt Lake City, UT
kwebb@cs.utah.edu

Sneha Kumar Kasera
School of Computing
University of Utah
Salt Lake City, UT
kasera@cs.utah.edu

Neal Patwari
McKelvey School of Engineering
Washington University in St. Louis
St. Louis, MO
npatwari@wustl.edu

Jacobus Van der Merwe
School of Computing
University of Utah
Salt Lake City, UT
kobus@cs.utah.edu

***Abstract*—Real world testbeds, like the POWDER platform (Platform for Open Wireless Data-driven Experimental Research), enable a broad range of mobile and wireless research. Given the flexibility of this platform, a key concern for platform users is selecting a set of wireless resources that will satisfy the requirements of their experiments. In this paper we present the design and implementation of WiMatch a wireless resource matchmaking system. We illustrate the utility of our approach by evaluating it in the POWDER platform.**

***Index Terms*—Mobile and Wireless testbed, Resource selection, Wireless measurements, Shout**

I. Introduction

Real world testbeds, like the POWDER platform (Platform for Open Wireless Data-driven Experimental Research) [2], allow for mobile and wireless research to be carried out in realistic conditions. Real radios, interference, and environmental conditions (e.g., weather, moving endpoints and obstructions) are features of this paradigm. Finding the right set of resources for conducting experiments in a wireless environment, such as the POWDER platform, is a difficult task. To put this into perspective, consider that such a platform includes dozens of radio resources with diverse paths. The more complex a user's desired experiment is, the more constraints they will need to satisfy simultaneously. Attributes such as frequency, bandwidth, link budget, and device capabilities may all need to satisfy particular requirements for a given setup. Even with knowledge of individual device attributes and pairwise link characteristics, how does a researcher put together the experiment they want to run? They may have a particular communication scenario in mind that they'd like to recreate. For example, they may want two rooftop base station radios and three endpoints, and the base stations overlap in RF coverage such that both can communicate with the endpoints. The communications channels should also have a particular minimum link budget. What if a fourth endpoint that is isolated (unable to receive signals) from the other clients is added?

How the researcher determines if this set of criteria can be met by the radio resources available is not clear. For simple requirements, it may be enough to show an interactive coverage map and let the researcher search to find what they need. As the number of resources and constraints grows, however, such manual inspection quickly becomes tedious and even intractable. In this paper we present our work on the WiMatch wireless resource matchmaking system. With WiMatch we demonstrate that it is possible to take succinct user-specified wireless requirements, and, based on real measurements, map them to resources in a fixed outdoor wireless system. We illustrate the practical utility of our approach by using WiMatch to select radio resources meeting user-specified requirements in the POWDER wireless platform.

The WiMatch system consists of three main components: (i) A query language for specifying requirements. (ii) Measurements over the radio environment. (iii) Mapping from the queries (requirements) to the resources (measurements).

Query Language We develop a targeted domain specific query language that makes wireless system inquiries precise and concise. This query language has inherent support for wireless environment aspects, e.g., link budgets, frequency, and bandwidth.

Wireless Measurements Rather than using propagation models to predict RF power levels, we take periodic empirical measurements in situ. We have developed the Shout measurement framework as part of WiMatch. Shout performs orchestrated wireless measurements across a distributed set of radios to empirically collect link budget estimates.

Mapping Queries to Resources The mapping algorithm is the bridge between user inquiries and system (environment and device) characteristics. It takes parsed user queries and converts them into codified requirements (constraints expressed as first order logic). These requirements are then mapped to measurement data using Microsoft Research's Z3 [4] satisfiability modulo theories (SMT) solver. The intuition behind this solver is that it uses approximation techniques to search a solution space that otherwise increases combinatorially in the number of variables being searched. The results are subsequently transformed into a response (set of matching resources) to send back to the user.

We make the following contributions: (i) We design and implement the WiMatch system to enable the selection of wireless resources in a real world mobile and wireless testbed based on user specified parameters. (ii) We evaluate WiMatch in the POWDER mobile and wireless platform. (iii) Our realization of WiMatch is available as open source enabling others to use and or extend its functionality.

II. Related Work

Expressing Wireless Requirements While testbeds typically have mechanisms for requesting particular devices, to the best of our knowledge, none provide a way to succinctly specify wireless requirements. There are, however, several domain-specific query languages that are peripherally related. One example is Spatial SQL [5] which operates over spatial databases. Since geospatial queries are part of what we consider here, we looked at how Spatial SQL and similar spatial query languages structure their queries. Most have SQL-like syntax with domain-specific grammar extensions. Languages for processing time series data, such as InfluxQL [10], also have SQL-like syntax with semantic extensions for interpreting and processing streaming data. Most of these works suggest that extending SQL syntax makes sense because users are already familiar with SQL, and SQL's semantics provide an intuitive natural language flow. For these reasons, we chose to use SQL as the basis for the WiMatch input language.

Finding Resources for Users Other testbeds have addressed the issue of guiding users to appropriate resources. In the Emulab testbed [16], the *assign* [12] algorithm maps from user-supplied network topologies to testbed resources (servers and network links) using a simulated annealing process. *Assign* is focused on wired network links and doesn't provide a particular query language interface. Additionally, *assign* was not designed to handle values within a range. Wireless factors such as signal strength, operating frequency and bandwidth, pairwise device relationships, and device mobility all require continuous range evaluations. In the ORBIT testbed [11], questions about the RF environment and device properties are addressed using a static set of data. It is up to the researcher to reason about these properties manually when selecting devices and interpreting communication performance.

Methods for measuring the wireless environment There is a large area of work covering how wireless systems behave where endpoints are modeled according to a random process [7], [9], [17]. In contrast, we study a predictable environment where link budgets are expected to remain stable, and the environment is 'clean' (i.e., the channel characteristics are known and transmissions are under the control of the experimenter). Several studies have looked at particular wireless deployments [1], [8], [13]. In our work, we are less interested in comparisons to path loss models, and more concerned with efficiently collecting real link characteristics for use in matching with user requirements.

III. WiMatch System Design

As shown in Figure 1, WiMatch is comprised of three high level components: A user-facing query language, measurements of the wireless channels between devices, and a mapping algorithm to put these together. The result is a system that allows users to find wireless resources that match their requirements. Next we cover the design of each component in detail.

A. Query Language

Domain specific languages (DSL) provide compact syntax to encode domain-specific concepts and provide mechanisms for expressing requirements such that a companion algorithm can find matching data. In this work, we develop a query DSL as part of WiMatch for specifying wireless communications concepts, from which requirements can be extracted and then mapped to radio device and RF environment measurement data.

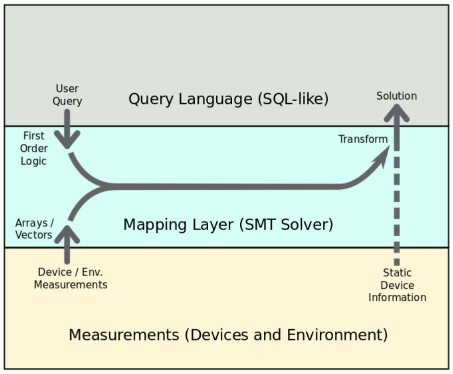

User Query
Query Language (SQL-like)
Solution
First Order Logic
Transform
Arrays / Vectors
Mapping Layer (SMT Solver)
Device / Env. Measurements
Static Device Information
Measurements (Devices and Environment)

Fig. 1: WiMatch Workflow

Communication range Wireless testbed users want to know if combinations of devices are within a tolerable link budget for communication. This link budget is a function of several wireless communication attributes such as center frequency, channel bandwidth, transmit power (gain), and received signal strength.

Radio types and capabilities Users need to know what devices are available along with their capabilities and limitations. Maximum channel bandwidth and tuning range are important pieces of information. Such device characteristics inform users of where they can operate and how. Another key performance metric is the maximum gain of the transmitters where output stops behaving linearly (P1dB point).

Overlapping device sets More than one or two devices in an experiment setup requires considering the interrelationships of these devices. Users will benefit from being able to ask if one set of devices can communicate with another set. This class of queries builds on the more basic notions of the individual device performance and capabilities already discussed.

1) Structure of queries: Given the classes of information we want to support in WiMatch queries, we move on to the structure of the language and how it supports these. (Note that WiMatch only supports search operations, as there is no need to support changes to stored data (insertion or modification).) To this end, we extend the canonical SQL search command, SELECT, and elide all other top-level commands found in SQL.

WiMatch extends SQL grammar with directives, types, and interpolation features that are either specific to, or useful in, wireless communication queries. The following example illustrates several of the salient features of WiMatch queries:

Listing 1: Language Example

```
SELECT 2 OF type1, 2 OF type2, 1 of type3 WHERE type1 IS A ’base_station’,
type2 IS AN ’endpoint’, frequency = 2625, bandwidth = 10,
LB(type1 <−> type2) > 20, LB(1 of type2 <−> type2) < 10;
```

This query asks for 2 devices of type1 and 3 of type2 where type1 devices are base stations and type2 devices are endpoints. The quoted string types indicate arbitrary tags that the devices have been labeled with. Declaring type tags like this limits the mapper to devices with matching tags. Tags can be omitted, with device characteristics used instead to match to appropriate resources. The query next specifies two global communication requirements: All devices, regardless of type, will operate at center frequency 2625 MHz and will use a channel bandwidth of 10 MHz. The query goes on to declare two set-based constraints. It first asks that the link budget ('LB') between all type1 and type2 devices be greater than 20 dB (above noise and interference). Again, units are implied by the property; decibels are assumed here since the quantity deals with signal strength. The double-ended arrow is an operator that indicates a two-way relationship. The one way (onto) set operator is also supported, allowing for asymmetric relationships. Finally, the query requests that one of the type2 devices be isolated from all other type2 devices via a low link budget.

There are a few other language features of note in this query. First, as seen in the LB clause, a lack of quantifier or qualifier for a type implies all devices of that type. Second, as seen in the $LB() < 10$ clause, WiMatch implicitly removes overlap between device sets used in the same clause, going from left to right. In the example, it doesn't make sense for the isolated type2 device to be in the type2 set listed after it (it can't be isolated from itself). This shorthand obviates the need for the query author to produce a verbose and awkward construction that declares the omission of overlapping set elements.

2) Language Features: Creating a feature rich query environment for wireless communication requires us to broadly consider the variety of questions users will want to ask. We have divided up the types of queries into two groupings: static and dynamic knowledge queries. The former considers properties that don't change or that infrequently change (on the order of months) over time. These include lists of devices, the set of available device types, individual device performance and calibration measurements. Dynamic data concerns relationships that are tracked on an ongoing basis as they are subject to change over time. Examples include relationships between devices (link budgets), and environmental factors (third-party interferers and weather).

B. Measurements

Some wireless related measurement data is static over long time frames, such as individual radio device performance. Other data is more dynamic. In a lab setting, we might think of inter-device performance characteristics as relatively static, but this assumption won't hold over longer time scales in an outdoor environment. New construction, demolitions, landscaping, weather, and other physical clutter dynamics will change RF propagation behavior. Therefore, there is a need for ongoing measurements of over-the-air signal propagation between devices. Important device and inter-device characteristics include: (i) **Device transceiver bandwidth.** The devices selected need to support the channel bandwidth needed by the user. This is a static device parameter that can be taken straight from a data sheet. (ii) **Device frequency range.** Clearly devices that satisfy a user's query must be able to operate in the frequency range they have specified. This is another static device parameter. (iii) **Device-to-device path loss.** Over-the-air signal loss between pairs of devices is necessary for calculating link budget. It is dependent on the dynamics of the outdoor environment and how these dynamics affect RF propagation.

We continue next with details on what individual device, inter-device, and environmental measurements we collect for supporting solutions to user resource queries.

1) Radio device performance characterization: Certain characteristics are inherent in the design of radio devices, and can simply be taken from data sheets. These include digital to analog converter (DAC) and analog to digital converter (ADC) resolution (bit width), gain step granularity, tunable frequency range, minimum and maximum bandwidth and bandwidth step granularity. These static characteristics are included in the datasets used by the WiMatch system.

2) Device-to-Device Communication Measurements: Collecting ongoing measurements that capture relationships between field deployed radio devices is important for the mapping process. When users ask questions related to link budgets between devices, time series data such as received signal strength at device A from transmissions originating from device B are needed. A measurement procedure is needed that iterates over all pairings of radio devices. This procedure should strive to both ensure the integrity of the measurements, and to reduce the overall time required. We have developed the Shout tool for performing such measurements. The basic procedure Shout uses is as follows.

For each device T in the list of all N devices, do:

1) Designate T as the transmitter.
2) Have all other devices act as receivers.
3) Sweep the allowed/available frequency range R in increments of i.
    a) T transmits for a fixed time t at power p.
    b) All receivers measure T's signal during time t.

The measurement procedure has run time complexity $O(N * t * R/i)$. As a concrete example, with 20 devices (N), a range of 400 MHz (R), a step size of 1 MHz (i), and measurement time of 2 seconds (t), a full data collection run would take about 4.5 hours. Shout takes two measurements at each frequency step: Without the transmitter active to gather noise power, and then with the transmitter to measure received power. These values are subtracted to get the "power over noise" value used in mapping link budget constraints.

C. Mapping

There is a large class of satisfiability problems that are not readily solved by exhaustive search. These include general boolean satisfiability and integer linear max/min problem classes. The solution space for such problems grows exponentially or even combinatorially with the size of the problem, which means that solutions to problems involving large numbers of constraints cannot be found on reasonable time scales on modern computer systems. As common as these problems are, "shortcut" algorithms that use approximation and other techniques abound. Dozens of satisfiability modulo theorem (SMT) solvers exist for finding solutions to constraint problems [3], [4]. Most of these take an expression in first order logic (FOL) and operate over it to find a solution.

Having developed a query language user interface and radio environment measurement procedures, we now develop the bridge between these mechanisms: a mapping algorithm. Our approach utilizes the Z3 satisfiability modulo theories (SMT) solver from Microsoft Research. Accommodating an SMT solver requires that we transform WiMatch queries and radio measurements into forms appropriate for Z3. Queries are converted into constraints expressed in first order logic (FOL) statements. Likewise, collected radio measurements are converted into matrices that are also used in these logic statements. The solver will attempt to find matching measurement data and device characteristics that satisfy the encoded requirements. If a solution is found, the devices that satisfy the query requirements are identified by the solver, and we pass these back to the user. If no satisfiable solution can be found, we inform the user of this.

1) Transforming The Query: The first step in the mapping procedure is to take the WiMatch query and convert it into a set of constraints. The query is first parsed into an abstract syntax tree (AST). This data structure encodes the essential parts of the query in a form that is amenable to searching, validation, annotation, and transformation. Using the AST, we check that the query is well-formed before proceeding further, aborting and informing the user if it isn't. The AST is then scanned for the measurement data that will be needed, which is then fetched. The validated and annotated AST is then transformed into Z3 FOL statements.

The last step taken by the transformer is to combine together all clauses and expand all pairwise relationships in the mapping statements (LB statements). Although Z3 supports quantifiers over sets, these do not support range specifications (they are meant to be used in proofs). This means that WiMatch must perform an expansion of all pairwise relationships expressed in the query. For example, the LB expression with the mapping relationship argument expands to pairwise clauses for each member of the sets specified in the argument.

An important observation about the above FOL statement is that clauses (LB statements) are dependent on the frequency of operation and channel bandwidth. The measurements collected by Shout for the particular frequency range requested in the query are extracted from the measurements dataset and used in the FOL expression. These measurements are averaged over the channel bandwidth, though other data reductions could be accommodated (min, max, quartiles, etc.). We can finally pass the fully expanded FOL statement to the Z3 SMT solver for analysis.

2) SMT Solver: The Z3 solver from Microsoft Research [4] was chosen for this work because it is open source, supports multiple programming languages, and, most importantly, it supports all satisfiability theories required to map WiMatch queries to underlying radio measurements.

Z3 takes the FOL statements, solves these (if possible), and outputs solutions for the variables in the form of integer identifiers. We do a straightforward mapping from these identifiers to the radio devices that they represent. The resulting solution is then formatted and sent back to the user. Multiple solutions to any given user request are possible, but the solver will stop at the first one found. Alternative solutions could be offered by including constraints that omit prior solutions and running the mapper again.

3) Transforming Measurement Data: The SMT solver requires measurement data inputs in order to perform the mapping from user requested characteristics specified in queries. Data for particular radio devices models and from measurement runs (see Section V-A) is used to form an adjacency array needed by the solver. Frequency and bandwidth parameters from the query are used as keys into the measured data to look up device characteristics and inter-device path measurements. Link budgets between devices are calculated across this range based on measured noise floor, and measured carrier wave receive power. These link budgets are collected into the dataset discussed in Section V-A. The most recent set of measurements collected are averaged. Records representing available devices are produced from static data for the set of devices in the testbed. These correspond to the 'base_station' and 'endpoint' terms in example Listing 1.

IV. Implementation

The implementation of our matchmaking and measurement framework can be broken into two essential parts: WiMatch, which consists of a query parser and constraint mapper, and the Shout distributed measurement tools. There is also some user-facing front-end logic that puts all of these components together. WiMatch and Shout are written entirely in Python. All of these components were designed with extensibility and modularity in mind; our overall implementation goal is to provide a framework and set of essential functionality. We fully expect that additional features can and will be added at all layers. Both the overall WiMatch system and the Shout tool are available as open source [14].

V. Evaluation

We evaluated WiMatch using the outdoor over-the-air resources available on the POWDER platform. We used POWDER cellular rooftop radios as transmitters, POWDER fixed endpoints as receivers. For conducting measurements, we designed a POWDER profile that allows us to request arbitrary sets of rooftop and fixed endpoint radios, as well as their corresponding compute resources. We instantiated the profile to gather the results in this paper on November 13, 2020. Our measurements targeted the downlink of LTE band 7, from 2620 - 2640 MHz. An important aspect of this range (at the time we did the measurements) is that the first 10 MHz is clear of 3rd party transmitters (no interference), while the latter 10 MHz includes an operating wireless incumbent (interference present).

A. *Component evaluation: Path RF Measurements*

To evaluate Shout's measurement component, we used its "all paths" measurement function to automatically collect measurements between rooftop transmitters and fixed endpoint receivers. Shout's "all paths" tool measures single continuous wave (CW) power over noise, sweeping across the range of frequencies in user-definable steps. At each frequency step, it first measures channel noise, and subsequently CW power at a that frequency (a narrow 10 KHz passband filter is employed in software). We had the tool gather data across the 2620 - 2640 MHz frequency range in 500 KHz steps. Additionally, we performed three runs for each frequency step to observe measurement variability.

We looked at the power of the individual frequency steps, as shown in Figure 2. This plot has sets of bars representing each 500 KHz step for every transmitter and receiver pair between 2620 - 2625 MHz. As can be noted, the power of individual steps is similar, but variations across the steps are present. This is most likely explained by subtle shifts in constructive and destructive multi-path at the receiver with respect to continuous wave frequency. Another contributing factor is the variability in the individual radio transmitter and receiver analog RF stages (amplification, filtering). This effect is apparent in the narrow spikes at the top of the bars, which shows the first standard deviation over the three runs taken at each frequency step.

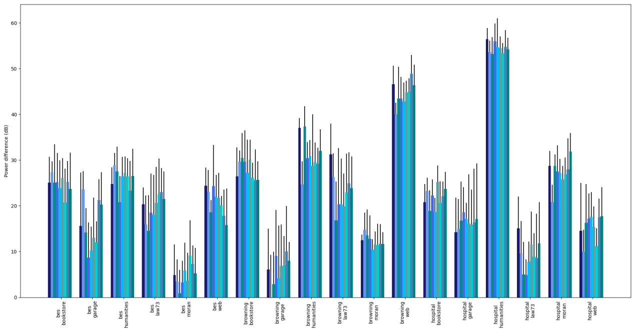


Fig. 2: Bar plot of 500 KHz steps for TX/RX pairs in Nov. 13 measurements - clear channel.

Figure 3 shows the same plot in the upper part of measured range, from 2635 - 2640 Mhz, where the 3rd party incumbent is operating. This figure shows lower power over noise values as a result of the incumbent's interference. Figure 4 shows two power spectral density plots computed using the same samples used to calculate power over noise. These plots show the 2.5 Mhz offset CW at 2620 Mhz and 2635 Mhz, respectively. As can be noted in the second plot, the incumbent signal intrudes on the continuous wave's power. This interference necessarily reduces the link budget that a user of POWDER would have when operating in this range. Our measurements capture this effect, and we demonstrate next how this affects which devices can satisfy a desired link budget during matchmaking further on in this section.

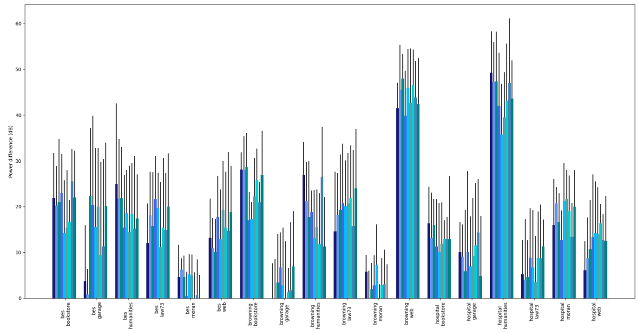
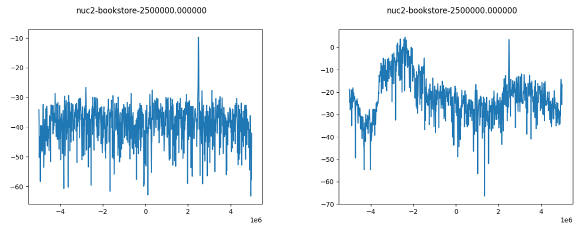


Fig. 3: Bar plot of 500 KHz steps for TX/RX pairs in Nov. 13 measurements - occupied channel.



(a) Clear channel

(b) Incumbent channel

Fig. 4: Clear vs. occupied channels in band 7 downlink

B. *Component evaluation: Query mapping*

We evaluated our approach with real radios and measured link budgets from the POWDER testbed. Using LTE band 7 downlink data, from Section V-A, we evaluated how WiMatch maps queries to real world measurements and devices. Table I shows the results of several query runs against the data. Note that only the parts of the query that change are shown in the table. A full query as used for producing the table is shown in Listing 2. A key observation is that, as the requested 10 MHz channel's center frequency moves from 2625 to 2635, the available link budget drops due to the 3rd party incumbent operating from 2630 MHz on up (Rows 1, 3 and 6).

Listing 2: Query Example

```
SELECT 2 of t1, 3 of t2 where t1 is a 'bs', t2 is an 'fe', frequency = 2625,
bandwidth = 10, LB(t1 −> t2) >= 20;
```

C. *End-to-End evaluation with srsLTE*

The srsLTE [6] software suite includes an end-to-end implementation of 4G LTE (largely 3GPP Release 14 compliant). It can be used to successfully communicate over the air on the POWDER outdoor wireless platform over several wireless channels between the devices. We explore how measured srsLTE radio access network (RAN) connectivity and performance correlate to the link budgets measured by Shout.

Row	Query	Result/Mapping
1	frequency = 2625, LB(t1 -> t2) >= 20	t1_1:bes, t1_2:browning, t2_1:bookstore, t2_2:web, t2_3:humanities
2	frequency = 2625, LB(t1 -> t2) >= 25	No mapping found
3	frequency = 2630, LB(t1 -> t2) >= 20	No mapping found
4	frequency = 2630, LB(t1 -> t2) >= 17	t1_1:browning, t1_2:bes, t2_1:humanities, t2_2:bookstore, t2_3:law73
5	frequency = 2635, LB(t1 -> t2) >= 17	No mapping found.
6	frequency = 2635, LB(t1 -> t2) >= 16	t1_1:browning, t1_2:bes t2_1:bookstore, t2_2:humanities, t2_3:law73

TABLE I: Sample queries and results

In this test, we have a single srsLTE instance and eNodeB instances for each rooftop transmitter measured in Section V-A. srsLTE UE software runs on the ‘nuc2‘ devices at each fixed endpoint listed in this same table. Individual device pairs (eNodeB and UE) are activated to isolate performance to the corresponding channel between them. We show a representative subset of the results in Table II. The corresponding link budgets measured by Shout are annotated in each table cell in parentheses. We observe that pairings with minimal connectivity tend to line up with Shout's measured link budgets above 20 dB, but below 30 dB. Good links occur between 30 to 40 dB, and strong links are seen above 40 dB. However, the correlation is not perfect. This can likely be explained by the weaker uplink power available on the fixed endpoint UE side as well as by individual device variations in power. Exploring the inconsistencies in correlation between these two data sets is future work. Despite these inconsistencies, link budget measurements from Shout generally trend with srsLTE performance.

	Behavioral	Browning	Friendship
Warnock	no (21)	9.4 (44)	no (< 10)
Bookstore	1.0 (22)	7.03 (26)	no (< 10)
Humanities	can't rach (25)	0.032 (33)	no (17)
Law73	no (23)	no (16)	no (13)
Moran	no (< 10)	no (12)	no (< 10)
Garage	no (15.8)	no (< 10)	0.689 (30)
Guesthouse	no (< 10)	no (< 10)	no (< 10)

TABLE II: srsLTE performance annotated with Shout measurements: Downlink throughput (MHz) and CW power over noise+interference (dB, in parenthesis).

VI. Conclusion

Our work on WiMatch has resulted in a functional and feature rich end-to-end tool that allows users to express RF requirements in intuitive SQL-like language. Driven by tools that measure the real RF environment, WiMatch allows these requirements to be mapped to resources that satisfy them. We have demonstrated the utility of this end-to-end matchmaking system through evaluation of the individual components, and through the comparison of real world application performance over matched resources. We have packaged our work into a POWDER profile that enables others to replicate our results and to serve as a starting point for related research efforts [15] and expect to continue to enhance WiMatch to perform new and different measurements and mapping functions.

Acknowledgements: This material is based upon work supported by the National Science Foundation under Grant Number 1827940.

References

[1] Andersen, J. B., Rappaport, T. S., and Yoshida, S. Propagation measurements and models for wireless communications channels. *IEEE Communications Magazine 33*, 1 (1995), 42–49.

[2] Breen, J., Buffmire, A., Duerig, J., Dutt, K., Eide, E., Hibler, M., Johnson, D., Kasera, S. K., Lewis, E., Maas, D., et al. Powder: Platform for open wireless data-driven experimental research. In *Proceedings of the 14th International Workshop on Wireless Network Testbeds, Experimental evaluation & Characterization* (2020), pp. 17–24.

[3] Bruttomesso, R., Pek, E., Sharygina, N., and Tsitovich, A. The opensmt solver. In *International Conference on Tools and Algorithms for the Construction and Analysis of Systems* (2010), Springer, pp. 150–153.

[4] De Moura, L., and Bjørner, N. Z3: An efficient smt solver. In *International conference on Tools and Algorithms for the Construction and Analysis of Systems* (2008), Springer, pp. 337–340.

[5] Egenhofer, M. J. Spatial sql: A query and presentation language. *IEEE Transactions on knowledge and data engineering 6*, 1 (1994), 86–95.

[6] Gomez-Miguelez, I., Garcia-Saavedra, A., Sutton, P. D., Serrano, P., Cano, C., and Leith, D. J. srslte: an open-source platform for lte evolution and experimentation. In *Proceedings of the Tenth ACM International Workshop on Wireless Network Testbeds, Experimental Evaluation, and Characterization* (2016), pp. 25–32.

[7] Ikuno, J. C., Wrulich, M., and Rupp, M. System level simulation of lte networks. In *2010 IEEE 71st Vehicular Technology Conference* (2010), IEEE, pp. 1–5.

[8] Imoize, A. L., Ibhaze, A. E., Nwosu, P. O., and Ajose, S. O. Determination of best-fit propagation models for pathloss prediction of a 4g lte network in suburban and urban areas of lagos, nigeria. *West Indian Journal of Engineering 41*, 2 (2019).

[9] Medard, M. The effect upon channel capacity in wireless communications of perfect and imperfect knowledge of the channel. *IEEE Transactions on Information theory 46*, 3 (2000), 933–946.

[10] Naqvi, S. N. Z., Yfantidou, S., and Zimányi, E. Time series databases and influxdb. *Studienarbeit, Université Libre de Bruxelles* (2017).

[11] Raychaudhuri, D., Seskar, I., Ott, M., Ganu, S., Ramachandran, K., Kremo, H., Siracusa, R., Liu, H., and Singh, M. Overview of the orbit radio grid testbed for evaluation of next-generation wireless network protocols. In *IEEE Wireless Communications and Networking Conference, 2005* (2005), vol. 3, IEEE, pp. 1664–1669.

[12] Ricci, R., Alfeld, C., and Lepreau, J. A solver for the network testbed mapping problem. *ACM SIGCOMM Computer Communication Review 33*, 2 (2003), 65–81.

[13] Sun, S., Rappaport, T. S., Rangan, S., Thomas, T. A., Ghosh, A., Kovacs, I. Z., Rodriguez, I., Koymen, O., Partyka, A., and Jarvelainen, J. Propagation path loss models for 5g urban micro- and macro-cellular scenarios. In *2016 IEEE 83rd Vehicular Technology Conference (VTC Spring)* (2016), IEEE, pp. 1–6.

[14] Webb, K. Shout measurements framework. https://gitlab.flux.utah.edu/powderrenewpublic/shout.git, 2020.

[15] Webb, K., and Van der Merwe, J. Powder profile - wimatch. https://www.powderwireless.net/p/PowderProfiles/WiMatch-OTA, 2021.

[16] White, B., Lepreau, J., Stoller, L., Ricci, R., Guruprasad, S., Newbold, M., Hibler, M., Barb, C., and Joglekar, A. An integrated experimental environment for distributed systems and networks. *ACM SIGOPS Operating Systems Review 36*, SI (2002), 255–270.

[17] Wu, D., and Negi, R. Effective capacity: a wireless link model for support of quality of service. *IEEE Transactions on wireless communications 2*, 4 (2003), 630–643.